

Re: Initial 6.1 questions

Source: <http://unix.derkeiler.com/Mailing-Lists/FreeBSD/performance/2006-06/msg00030.html>

- *From:* "Kip Macy" <kip.macy@xxxxxxxxxx>
 - *Date:* Tue, 13 Jun 2006 12:34:44 -0700
-

I have a number of issues with our current locking regime and our propensity for disabling interrupts. I have in mind some ideas for reducing interrupt disabling and eliminating scheduling contention except in the case of one cpu stealing a thread from another cpu's runqueue. I'll try to dash that off early this evening. This should also greatly reduce the overhead of timer interrupts.

-Kip

On 6/13/06, Robert Watson <rwatson@xxxxxxxxxxxxx> wrote:

On Tue, 13 Jun 2006, David Xu wrote:

> On Tuesday 13 June 2006 04:32, Kris Kennaway wrote:
>> On Mon, Jun 12, 2006 at 09:08:12PM +0100, Robert Watson wrote:
>>> On Mon, 12 Jun 2006, Scott Long wrote:
>>>> I run a number of high-load production systems that do a lot of network
>>>> and filesystem activity, all with HZ set to 100. It has also been shown
>>>> in the past that certain things in the network area were not fixed to
>>>> deal with a high HZ value, so it's possible that it's even more
>>>> stable/reliable with an HZ value of 100.
>>>>
>>>> My personal opinion is that HZ should go back down to 100 in 7-CURRENT
>>>> immediately, and only be incremented back up when/if it's proven to be
>>>> the right thing to do. And, I say that as someone who (errantly) pushed
>>>> for the increase to 1000 several years ago.
>>>>
>>>> I think it's probably a good idea to do it sooner rather than later. It
>>>> may slightly negatively impact some services that rely on frequent timers
>>>> to do things like retransmit timing and the like. But I haven't done any
>>>> measurements.
>>>>
>>>> As you know, but for the benefit of the list, restoring HZ=100 is often an
>>>> important performance tweak on SMP systems with many CPUs because of all
>>>> the sched_lock activity from statclock/hardclock, which scales with HZ and
>>>> NCPUS.
>>>>
>>>> sched_lock is another big bottleneck, since if you 32 CPUs, in theory you

Re: Initial 6.1 questions

- > have 32X context switch speed, but now it still has only 1X speed, and there
- > are code abusing sched_lock, the M:N bits dynamically inserts a thread into
- > thread list at context switch time, this is a bug, this causes thread list
- > in a proc has to be protected by scheduler lock, and delivering a signal to
- > process has to hold scheduler lock and find a thread, if the proc has many
- > threads, this will introduce long scheduler latency, a proc lock is not
- > enough to find a thread, this is a bug, there are other code abusing
- > scheduler lock which really can use its own lock.

I've added Kip Macy to the CC, who is working with a patch for Sun4v that eliminates sched_lock. Maybe he can comment some more on this thread?

Robert N M Watson
Computer Laboratory
Universty of Cambridge

freebsd-performance@xxxxxxxxxxx mailing list

<http://lists.freebsd.org/mailman/listinfo/freebsd-performance>

To unsubscribe, send any mail to "freebsd-performance-unsubscribe@xxxxxxxxxxx"